Programming Assignment – 5

Circus of Plates

Prepared by:

Aya Sameh Esmail (1)

Salma Hesham Mohammed Ragab (34)

- ## User guide:

  If you're the type of person whose reactions are fast, then this game is definitely for you!

  Circus of Plates depends on how fast you go and catch the shapes falling from a mechanical belt above you.

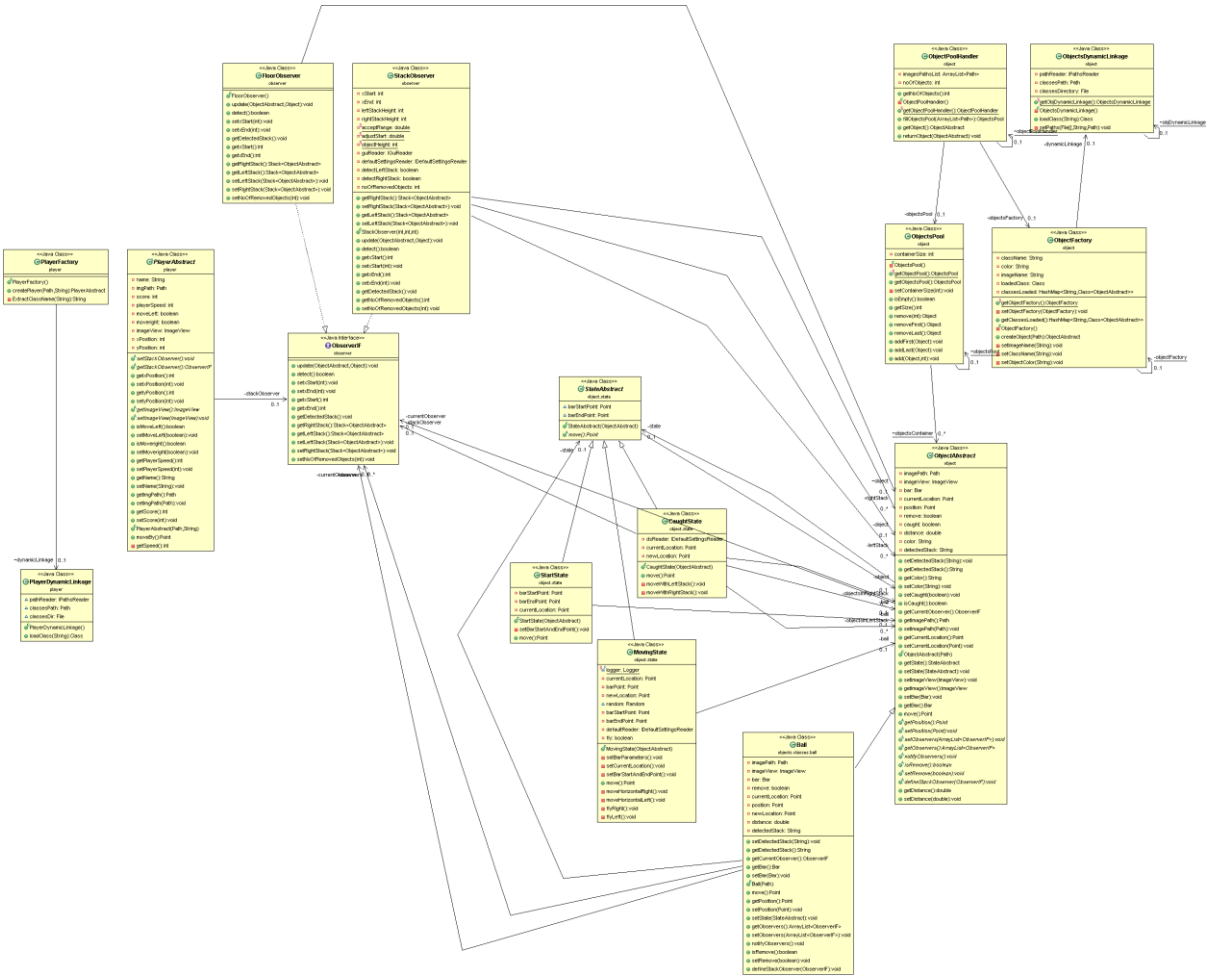
  How to play:

  - Wait for the game to load. Everything comes with a price.
  - If you're new to this then you should first choose "Start New Game", if not then load your saved game and increase your scoring!
  - Each user should then choose the player he/she wants to play with, and set their name.
  - The game supports up to 4 players.
  - Now comes the interesting part! The DIFFICULTY! :D
  - There are three levels of difficulties in this version of the game.
    - → Easy Mode (Beginners Mode): You score a point once you catch three objects of the same color, regardless their shapes.
    - → Medium Mode (Intermediates Mode): A point is added to your score once you catch three identical objects (i.e having the same color and shape).
    - → Hard Mode (Experts Mode): You have to catch 4 similar objects; so that a point is added to your score!
  - Each user will control his/her character using some keyboard buttons.
  - The game depends on a timer, once it ends, the winner is announced! :D
  - Being busy won't make it hard to play this game, we don't want you to miss the fun here! A "Pause" key, can help you with this.
  - Finally, save your game so you can increase your score later on.

- ## Design patterns:
  1. Dynamic Linkage: Loads the classes dynamically at the beginning of the game
  2. Factory: It was used in create objects and players.
  3. Object Pool: It was used in putting a limit to the objects created in the scene.
  4. Singleton: It was used in most of the classes; to make sure there is only one instance of them.
     Most significant use was in the object pool class; so that only one object pool is available during the game.
  5. Observer: It observes the change which happens to the object, either if it is on floor or it is caught by one of the players.
  6. State: Detects the state of the object either it it's a starting state, moving state or caught.
  7. Strategy: Implements different calculating score strategy, easy, medium or hard.
  8. Iterator: Implemented by the object pool to know if it still has objects or it reached the empty state.
  9. Snapshot: (Memento) Saves the state of the object and the players (The Game) to use them in the saving and loading operation.
  10. MVC: Used to differentiate between the different parts of the design (Model – View – Controller).

- ## Design decisions and Assumptions:
  1. All paths, default settings or locations in the gui class are read from a configuration file.
  2. Most of the classes are dynamically loaded on loading for the first time.

- **UML Diagram:**



- **Sequence Diagram:**